What is claimed is:

Sub a

1. A decoder for determining the log likelihood logarithmically expressing the probability of passing a given state on the basis of the received value regarded as soft-input and decoding the input by using the log likelihood, said decoder comprising:

a linear approximation means for calculating a correction term to be added to the log likelihood, the correction term being expressed in a one-dimensional function relative to a variable; and

said linear approximation means being adapted to compute said correction term using a coefficient representing the gradient of said function for multiplying said variable, the coefficient being expressed as a power exponent of 2.

2. The decoder according to claim 1, wherein said linear approximation means discards lower bits of the input data according to the power exponent expressing the coefficient representing the gradient of said function.

3. The decoder according to claim 2, wherein said linear approximation means discards the bits from the lowest bit to the k-th lowest bit of the input data when the coefficient representing the gradient of said function is expressed by -2^{-k}.

4. The decoder according to claim 1, wherein said linear approximation means computes said correction term using the coefficient representing the gradient of the

Sub a2

intercept of said function, the coefficient being expressed as a power exponent of 2.

5. The decoder according to claim 4, wherein said linear approximation means computes said correction term using the coefficient representing the intercept of said function, the coefficient being expressed as $2^m - 1$.

6. The decoder according to claim 5, wherein said linear approximation means discards the bits from the lowest bit to the k-th lowest bit of the n-bit input data and inverts the m bits from the k-th lowest bit to the m+k-th lowest bit when the coefficient representing the gradient of said function is expressed by means of -2^{-k}.

7. The decoder according to claim 1, wherein said correction term shows a positive value.

8. The decoder according to claim 7, wherein said linear approximation means makes the correction term equal to 0 when a negative value is produced by computing said correction term.

9. The decoder according to claim 1, wherein said log likelihood is a log expression of said probability, using the natural logarithm.

10. The decoder according to claim 1, further comprising:

a first probability computing means for computing for each received value a first log likelihood logarithmically expressing a first probability determined by the code output pattern and said received value;

Sub a2

a second probability computing means for computing for each received value a second log likelihood logarithmically expressing a second probability of getting to each state from the coding starting state in the time series;

a third probability computing means for computing for each received value a third log likelihood logarithmically expressing a third probability of getting to each state from the coding terminating state in the inverted time series; and

said second probability computing means and said third probability computing means having the linear approximation means.

11. The decoder according to claim 10, further comprising:

a soft-output calculation means for calculating the log soft-output logarithmically expressing the soft-output in each time slot, using said first log likelihood, said second log likelihood and said third log likelihood.

12. The decoder according to claim 11, wherein said log soft-output is a logarithmic expression of said soft-output, using the natural logarithm.

13. The decoder according to claim 1, wherein said log likelihood is determined by computations replacing the multiplications for computing the probability by logarithmic additions and the additions for computing the probability by logarithmic maximum value computations and computations of said function.

14. The decoder according to claim 13, wherein a maximum a posteriori

Sub a2

probability decoding operation is conducted on the basis of the Log-BCJR algorithm.

15. The decoder according to claim 1, wherein convolutional codes are decoded.

16. A decoding method for determining the log likelihood logarithmically expressing the probability of passing a given state on the basis of the received value regarded as soft-input and decoding the input by using the log likelihood, said decoding method comprising:

a linear approximation step for calculating a correction term to be added to the log likelihood, the correction term being expressed in a one-dimensional function relative to a variable; and

said linear approximation step being adapted to compute said correction term using a coefficient representing the gradient of said function for multiplying said variable, the coefficient being expressed as a power exponent of 2.

17. The decoding method according to claim 16, wherein said linear approximation step is adapted to discard lower bits of the input data according to the power exponent expressing the coefficient representing the gradient of said function.

18. The decoding method according to claim 17, wherein said linear approximation step is adapted to discard the bits from the lowest bit to the k-th

Sub a2

lowest bit of the input data when the coefficient representing the gradient of said function is expressed by -2^{-k}.

19. The decoding method according to claim 16, wherein said linear approximation step is adapted to compute said correction term using the coefficient representing the intercept of said function, the coefficient being expressed as a power exponent of 2.

20. The decoding method according to claim 19, wherein said linear approximation step is adapted to compute said correction term using the coefficient representing the intercept of said function, the coefficient being expressed as $2^m - 1$.

21. The decoding method according to claim 20, wherein said linear approximation step is adapted to discard the bits from the lowest bit to the k-th lowest bit of the n-bit input data and inverts the m bits from the k-th lowest bit to the m+k-th lowest bit when the coefficient representing the gradient of said function is expressed by means of -2^{-k}.

22. The decoding method according to claim 16, wherein said correction term shows a positive value.

23. The decoding method according to claim 22, wherein said linear approximation step is adapted to make the correction term equal to 0 when a negative value is produced by computing said correction term.

24. The decoding method according to claim 16, wherein said log

Sub a2

likelihood is a log expression of said probability, using the natural logarithm.

25. The decoding method according to claim 16, further comprising:

a first probability computing step for computing for each received value a first log likelihood logarithmically expressing a first probability determined by the code output pattern and said received value;

a second probability computing step for computing for each received value a second log likelihood logarithmically expressing a second probability of getting to each state from the coding starting state in the time series;

a third probability computing step for computing for each received value a third log likelihood logarithmically expressing a third probability of getting to each state from the coding terminating state in the inverted time series; and

said second probability computing step and said third probability computing step including the linear approximation steps.

26. The decoding method according to claim 25, further comprising:

a soft-output calculating step for calculating a log soft-output logarithmically expressing the soft-output in each time slot, using said first log likelihood, said second log likelihood and said third log likelihood.

27. The decoding method according to claim 26, wherein said log soft-output is a logarithmic expression of said soft-output, using the natural logarithm.

28. The decoding method according to claim 16, wherein said log

Sub a2

likelihood is determined by computations replacing the multiplications for computing the probability by logarithmic additions and the additions for computing the probability by logarithmic maximum value computations and computations of said function.

29. The decoding method according to claim 28, wherein a maximum a posteriori probability decoding operation is conducted on the basis of the Log-BCJR algorithm.

30. The decoding method according to claim 16, wherein convolutional codes are decoded.